METHOD FOR MAPPING SCREEN DISPLAY OBJECTS WITH SPATIAL COORDINATES INTO AN HTML TABLE

FIELD OF THE INVENTION

The present invention concerns the field of information display, and more particularly a method for mapping screen display objects having spatial coordinates into an HTML table, so that the objects may be displayed conveniently by a wide range of applications such as web browsers.

BACKGROUND

As the Internet and World Wide Web become ever more widely accepted, designers who organize information for visual display increasingly approach the layout of screen display objects with certain predispositions favorable to standard Internet technology. In particular, a standard approach for specifying the locations of screen display objects on a display screen has evolved – each screen display object is enclosed by a rectangle, and the location of the screen display object is specified by spatial coordinates of the rectangle relative to the display screen.

The task of translating fixed position content with spatial coordinates to DHTML layered content is straightforward when a designer uses an advanced web-based language such as Dynamic HyperText Markup Language (DHTML). Unfortunately, this kind of translation has several disadvantages and limitations. One limitation is that DHTML is not universally accessible to all kinds of display terminals. For example, wireless devices such as hand-held communication terminals generally cannot read DHTML. Moreover, accessibility may be hampered by the complex DHTML layering. Also, consistency is lacking, as a DHTML browser may operate differently on two different platforms, and DHTML content may differ significantly between versions of a browser.

Thus there is a need for transforming the spatial coordinates of fixed-position screen display objects into a form that wireless devices can access without heroic manipulations, a form that provides consistency so a wide range of web browsers may be successfully used, and a form that provides universality so that different DHTML code need not be written for each browser version.

SUMMARY

The present invention provides a method of mapping screen display objects into a Hypertext Markup Language (HTML) table. Spatial coordinates for each screen display object are determined, and an HTML table having rows and columns is created, wherein row heights and column widths are determined by the spatial coordinates. The screen display objects are then loaded into selected cells of the HTML table for display, wherein selection of the cells is determined as a function of the spatial coordinates.

In addition to solving the aforementioned problems of accessibility, consistency, and universality, the present invention also provides a method for converting pre-existing content into flat HTML, and enables accessibility tools such as screen readers to read the resulting page content with less operational complexity. These and other aspects of the invention will be appreciated more fully when considered in the light of the following detailed description and drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagram that shows two exemplary screen display objects and associated spatial coordinates.

FIG. 2 is a flow chart that illustrates conversion of fixed-position content with spatial coordinates into an HTML table.

FIG. 3 is a flow chart that illustrates a way of determining a cell in the HTML table of FIG. 2 appropriate for holding a screen display object.

FIG. 4 shows an HTML table holding the two exemplary screen display objects of FIG. 1.

DETAILED DESCRIPTION

The present invention provides a method for mapping screen display objects with spatial coordinates to an HTML table, so that the screen display objects may be processed and displayed conveniently, consistently, and universally by a wide variety of applications such as web browsers.

A screen display object may be contained within an area of a display screen specified by spatial coordinates. For example, a screen display object may be contained within a rectangular area of the screen, where the four corners of the rectangle are specified by Cartesian coordinates. Such coordinates may also be called rectangular coordinates or x-y coordinates, and may be conveyed by ordered pairs with respect to a reference called the origin. For purposes of clear explanation, a running example of this kind is discussed below, although the present invention is, of course, not limited by the running example. As shown in FIG. 1, the example considers two exemplary screen display objects, a first screen display object 100 having spatial coordinates (2,1), (4,1), (4,6) and (2,6), and a second screen display object 110 having spatial coordinates (7,5), (11,5), (11,9) and (7,9), all with respect to an origin that has spatial coordinates (0,0).

FIG. 2 is a flow chart that shows the operation of an exemplary embodiment of a method according to the invention. As shown in FIG. 2, the spatial coordinates of the screen display objects are determined, to provide a first x coordinate, a second x coordinate, a first y coordinate, and a second y coordinate for each screen display object (step 200). Coordinates are ordered so that the numerical value of the first x coordinate is less than the numerical value of the second x coordinate, and the numerical value of the first y coordinate is less than the numerical value of the second y coordinate.

For example, the location of a rectangle may be described by the four ordered pairs (*a*,*c*), (*b*,*c*), (*b*,*d*), and (*a*,*d*). For this rectangle, the first x coordinate would be *a*, the second x coordinate would be *b*, the first y coordinate would be *c*, and the second y coordinate would be *d*. In particular, for the first screen display object 100 in the running example, the first and second x coordinates are 2 and 4, respectively, and the first and second y coordinates are 1 and 6, respectively. For the second screen display object 110 in the running example, the first and second x coordinates are 7 and 11, respectively, and the first and second y coordinates are 5 and 9, respectively.

A set of x coordinates and a set of y coordinates are created (step 205). The set of x coordinates includes the first and second x coordinates of each screen display object; likewise, the set of y coordinates includes the first and second y coordinates of each screen display object. Also, the x coordinate of the origin and the y coordinate of the origin are included in the set of x coordinates and the set of y coordinates, respectively (step 210). Duplicates of any element in the set of x coordinates are dropped, and duplicates of any element in the set of y coordinates are dropped. The remaining elements of each set are then sorted into numerical order. In the running example, the resulting set of x coordinates would be (0, 2, 4, 7, 11) and the set of y coordinates would be (0, 1, 5, 6, 9).

A count is then made of the numbers of elements remaining in the sets after duplicate elements are dropped (step 220). The number of elements in the set of x coordinates and the number of elements in set of y coordinates are denoted here by the variables M and N, respectively. In the running example, both M and N have the value 5.

An HTML table with M columns and N rows is then created (step 230). In the running example, the table would have five columns and five rows.

For each screen display object, a cell of the HTML table is identified (step 240). A cell includes the intersection of at least one column and at least one row of the table. When a cell includes more than one column, the included columns are contiguous in the table; likewise, when a cell includes more than one row, the included rows are contiguous in the table. For example, a cell might be the intersection of the second column with the third row of the table, or the intersection of the second column with the third and fourth rows of the table, or the intersection of the first and second columns with the fourth and fifth rows of the table, and so forth, but not the intersection of the first and third columns with the second row.

One way to identify the cells, which is shown in FIG. 3, is as follows: For a given screen display object, count the number of elements in the set of x coordinates that are less than or equal to the first x coordinate of the screen display object (step 300). The result identifies the initial column of an intersection that determines the cell. Count the number of elements in the set of x coordinates that are greater than the first x coordinate of the screen display object and less than or equal to the second x coordinate of the screen display object (step 310). The result indicates the number of contiguous columns, including the initial column, in the intersection. Count the number of elements in the set of y coordinates that are less than or equal to the first y coordinate of the screen display object (step 320). The result identifies the initial row of the intersection. Count the number of elements in the set of y coordinates that are greater than the first y coordinate of the screen display object and less than or equal to the second y coordinate of the screen display object (step 330). The result indicates the number of contiguous rows, including the initial row, in the intersection. The cell for the given screen display object is the intersection just described.

For the first exemplary screen display object 100, the first x coordinate is 2 and the second x coordinate is 4. The set of x coordinates, (0, 2, 4, 7, 11), includes two elements less than or equal to 2; thus, the initial column of the intersection is column 2. The set of x coordinates includes one element greater than 2 and less than or equal to 4; thus, the intersection includes just one column, which is the initial column, i.e., column 2. The first y coordinate is 1 and the second y coordinate is 6. The set of y coordinates, (0, 1, 5, 6, 9), includes two elements less than or equal to 1; thus, the initial row of the intersection is row 2. The set of y coordinates includes two elements greater than 1 and less than or equal to 6; thus, the intersection includes two rows, row 2 (i.e., the initial row) and row 3. Consequently, the cell for the first exemplary screen display object 100 is the intersection of column 2 with rows 2 and 3.

For the second exemplary screen display object 110, the first x coordinate is 7 and the second x coordinate is 11. The set of x coordinates, (0, 2, 4, 7, 11), includes four elements less than or equal to 7; thus, the initial column of the intersection is column 4. The set of x coordinates includes one element greater than 7 and less than or equal to 11; thus, the intersection includes just one column, which is column 4. The first y coordinate is 5 and the second y coordinate is 9. The set of y coordinates, (0, 1, 5, 6, 9), includes three elements less than or equal to 5; thus, the initial row of the intersection is row 3. The set of y coordinates includes two elements greater than 5 and less than or equal to 9; thus, the intersection includes two rows, row 3 (i.e., the initial row) and row 4. Consequently, the cell for the second exemplary screen display object 110 is the intersection of column 4 with rows 3 and 4.

Based on the set of x coordinates, the width of each column of the HTML table is determined, and, based on the set of y coordinates, the height of each row is determined (step 250 of FIG. 2). Each screen display object is then loaded into its appropriate cell (step 260).

One way to determine the widths of the columns of the table and the heights of the rows is to set the widths equal to the differences between consecutive elements in the set of x coordinates, and to set the heights equal to the differences between consecutive elements in the set of y coordinates. Thus, the width of the first column of the HTML table may be the difference found by subtracting the first element of the set of x coordinates from the second element; the width of the second column may be found by subtracting the second element of the set of x coordinates from the third element, and so forth, and likewise in kind for the heights of the rows, based on differences between consecutive elements in the set of y coordinates. By default, the final column and the final row of the table each have width and length of one, to provide a border.

In the running example, the set of x coordinates is (0, 2, 4, 7, 11) and the set of y coordinates is

(0, 1, 5, 6, 9). Consequently, the width of the first column would be 2, the width of the second column would be 2, the width of the third column would be 3, the width of the fourth column would be 4, and the width of the fifth column would be 1. Likewise, the height of the first row would be 1, the height of the second row would be 4, the height of the third row would be 1, the height of the fourth row would be 3, and the height of the fifth row would be 1. Such an exemplary table 400 is shown in FIG 4, with the exemplary screen display objects 100, 110 located in the appropriate cells.

From the foregoing description, those skilled in the art will appreciate that the present invention provides a versatile, consistent, and universal way of mapping fixed-position content screen display objects with spatial coordinates into an HTML table. The foregoing description is illustrative rather than limiting, however, and the present invention is limited only by the claims that follow.